

## **REMARKS**

The Office Action of August 10, 2006 has been reviewed and the comments therein were carefully considered. Claims 1-17 are pending. Claims 1-17 stand rejected.

### **Information Disclosure Statement**

The Office Action indicated that the Information Disclosure Statement filed on 8/3/03 contained copies of non-patent literature that were not legible and therefore placed in the file but not considered. Applicants are including additional copies of those documents along with a new IDS form for consideration by the Examiner.

### **Claim Rejections Under 35 USC §102**

Claims 1-17 are rejected under 35 USC §102(e) as being clearly anticipated by Kasichainula, et al., U.S. Patent No. 6,941,561. Applicants respectfully traverse the rejections.

Kasichainula discloses a method of distributing a program written using object orientated programming so that portions of the written program may be executed on more than one computer across a network. Two proxy objects are generated dynamically to enable communication over the intervening network by intercepting calls. The proxy objects allow method calls written for local invocation to be invoked over a network. The two proxies cooperate to hide the fact that the objects actually reside on different machines from the programmer, thereby sparing the programmer any need to be aware of the distributed nature of the system when writing his code.

Independent claims 1 and 13 include the claimed feature of “wrapping a reference to a second object within a second context with a proxy wrapper.” The Office Action points to the title and abstract of Kasichainula to show Applicants this claimed feature. Applicants respectfully disagree as neither the title of Kasichainula nor abstract disclose this claimed feature. The title of the Kasichainula patent is “Method and apparatus for remotely running objects using data streams and/or complex parameters” and the abstract states:

Proxy data stream handling and complex object parameter handling allow object orientated programs to be run as distributed programs without any explicit networking code, and without using an interface definition language (IDL). Two proxies are generated dynamically that allow method calls written for local invocation to be invoked over a network. These dynamically-generated proxies

allow calls to flow across a network as if they were local, and contain support for using data stream and complex objects as parameters.

Applicants respectfully submit that neither the title nor abstract of Kasichainula disclose or suggest Applicants claimed feature of “wrapping a reference to a second object within a second context with a proxy wrapper. (Emphasis added). The Office Action apparently is equating the creation of two proxy objects with the claimed feature of “wrapping a reference to a second object within a second content with a proxy wrapper.” (Emphasis added). Applicants respectfully submit that a proxy wrapper which includes a reference can not properly be equated with the creation of the two proxy objects of Kasichainula. Therefore, for at least this reason, Applicants respectfully submit that independent claims 1 and 13 are in condition for allowance. Dependent claims 2-5 and 14-17 are in condition for allowance for at least the same reason as independent claims 1 and 16 from which they ultimately depend.

Independent claim 6 includes the claimed feature of “at least one first object communicates with any of the at least one second object via indirect references wrapped in proxy wrappers.” (Emphasis added). Applicants respectfully submit that a proxy wrapper which includes indirect references can not properly be equated with the creation of the two proxy objects of Kasichainula. Therefore, for at least this reason Applicants respectfully submit that independent claim 6 is in condition for allowance. Dependent claims 7-11 which ultimately depend from independent claim 6 are in condition for allowance for at least the same reason as independent claim 6.

Independent claim 12 includes the claimed feature of “at least one agile object . . . in that the at least one agile object have no permanent context . . .” With respect to independent claim 12, the Office Action does not indicate where in Kasichainula Applicants claimed at least one agile object may be found in Kasichainula. With respect to dependent claim 3, the Office Action notes that “objects may be machine specific, which implies that they do not have to be and therefore wrapping would not be required.” (Office Action, page 3-4). Applicants respectfully submit that machine specific objects may not be equated with agile objects as machine specific objects contain the specific context of the specific machine. Therefore, for at least this reason, Applicants respectfully submit that independent claim 12 is in condition for allowance.

Applicants respectfully request consideration of the pending claims and a finding of their allowability. A notice to this effect is respectfully requested. Please feel free to contact the undersigned should any questions arise with respect to this case that may be addressed by telephone.

Respectfully submitted,

Dated: November 10, 2006

By: William J. Allen  
William J. Allen  
Registration No. 51,393  
Banner & Witcoff, Ltd.  
10 South Wacker Drive  
Suite 3000  
Chicago, IL 60606  
Tel: 312-463-5000  
Fax: 312-463-5001